JIRSS (2011)
Vol. 10, No. 2, pp 125-140

Penalized Bregman Divergence Estimation via Coordinate Descent

Chunming Zhang, Zhengjun Zhang, Yi Chai

Department of Statistics, University of Wisconsin-Madison, USA.

Abstract. Variable selection via penalized estimation is appealing for dimension reduction. For penalized linear regression, Efron, *et al.* (2004) introduced the LARS algorithm. Recently, the coordinate descent (CD) algorithm was developed by Friedman, *et al.* (2007) for penalized linear regression and penalized logistic regression and was shown to gain computational superiority. This paper explores the CD algorithm to penalized Bregman divergence (BD) estimation for a broader class of models, including not only the generalized linear model, which has been well studied in the literature on penalization, but also the quasi-likelihood model, which has been less developed. Simulation study and real data application illustrate the performances of the CD and LARS algorithms in regression estimation, variable selection and classification procedure when the number of explanatory variables is large in comparison to the sample size.

Keywords. Bregman divergence, LARS algorithm, quasi-likelihood, sparsity, Taylor expansion, variable selection.

MSC: Primary 62F12, 62F30; Secondary 62F05, 62J07.

Chunming Zhang (✉)(cmzhang@stat.wisc.edu), Zhengjun Zhang (zjz@stat.wisc.edu), Yi Chai(chaiyi@stat.wisc.edu)
Received: March, 2011; Accepted: June, 2011

1 Introduction

Technological invention and information advancement have revolutionized scientific research and technological development. Many sophisticated large-scale data sets have recently been collected, such as fMRI brain images, microarrays, protemics, large-scale surveys, financial data, and functional data. These new data sets and streams pose numerous challenges to conventional statistical or data mining methods due to not only the massive size, but also the large dimensionality.

Regularization or penalization is a technique aiming at obtaining well behaved solutions to overparameterized estimation problems. Model selection or variable selection, via penalization, is an appealing approach for selecting significant variables and removing irrelevant ones, thus achieving dimension reduction. Examples include the bridge regression (Frank and Friedman, 1993), the nonnegative garrote (Breiman, 1995), the least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996), SCAD (Fan, 1997; Fan and Li, 2001), and least angle regression (LARS) (Efron, *et al.*, 2004). The literature in this area is comprehensive and there have been significant theoretical developments. A review can be found in Bickel and Li (2001), Zhang, *et al.* (2010) and references therein.

Computationally, the original LARS algorithm (Efron, *et al.*, 2004) for L_1-penalization was developed for linear model estimation using the quadratic loss. Rosset and Zhu (2007) further studied the piecewise linear regularized solution paths for differentiable and piecewise quadratic loss functions with L_1 penalty. Recently, the coordinate descent (CD) algorithm was developed by Friedman, *et al.* (2007), Wu and Lange (2008), and Friedman, *et al.* (2010) for penalized linear regression and penalized logistic regression and was shown to gain computational superiority. It is desirable to explore the extent to which penalization methods using other types of loss functions can potentially benefit from the efficient CD and LARS algorithms.

To broaden the scope of penalization methods, this paper investigates the application of the CD algorithm to penalized Bregman divergence (BD) estimation for a wider class of models, including not only the generalized linear model, which has been well studied in the literature on penalized regression methods, but also the quasi-likelihood model, which has been less developed. The development is expected to be useful for other models. Furthermore, incorporating the weighted-L_1 penalty into the CD algorithm allows extensions to certain nonconvex penalties, such as the SCAD.

The rest of the paper is organized as follows. Section 2 reviews the CD algorithm for penalized linear regression with the L_1 penalty. Section 3 develops the CD algorithm for penalized BD estimation using the weighted-L_1 penalty. Section 4 evaluates the performances of the CD and LARS algorithms in penalized quasi-likelihood estimation via simulation study. Section 5 illustrates the penalized logistic regression with real data applications.

2 Coordinate Descent for Penalized Linear Regression

In this section, we will first briefly overview the CD algorithm for a linear regression model,

$$Y_i = \beta_0 + \boldsymbol{X}_i^T \boldsymbol{\beta} + \epsilon_i, \qquad i = 1, \ldots, n, \tag{2.1}$$

where β_0 and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ are unknown regression coefficients, $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip})^T$ is a p-variate predictor vector, Y_i is a scalar response variable, and the error ϵ_i satisfies $E(\epsilon_i \mid \boldsymbol{X}_i) = 0$. The penalized weighted least-squares estimation of $(\beta_0, \boldsymbol{\beta})$, using the L_1 penalty, amounts to solving the optimization problem,

$$\arg \min_{\beta_0, \beta_1, \ldots, \beta_p} \left\{ \frac{1}{2} \sum_{i=1}^{n} w_i (Y_i - \beta_0 - \boldsymbol{X}_i^T \boldsymbol{\beta})^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.2}$$

where $\{w_i\}$ are non-negative weights associated with the part of quadratic loss functions, and λ_n is a positive tuning constant governing the amount of penalization. The conventional Lasso penalized linear regression estimation,

$$\arg \min_{\beta_0, \beta_1, \ldots, \beta_p} \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - \beta_0 - \boldsymbol{X}_i^T \boldsymbol{\beta})^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| \right\},$$

corresponds to the use of $w_i \equiv 2/n$ or other non-negative constants.

Computationally, the CD algorithm solves (2.2) in an iterative way. Suppose we are given some initial estimates $\{\widetilde{\beta}_k\}_{k=0}^p$ of $\{\beta_k\}_{k=0}^p$. Denote by $\widetilde{Y}_i = \widetilde{\beta}_0 + \sum_{k=1}^p X_{ik} \widetilde{\beta}_k$ the fitted responses and by $\widetilde{r}_i = Y_i - \widetilde{Y}_i$, $i = 1, \ldots, n$, the residuals. In the CD algorithm, for each coordinate index $j = 1, \ldots, p$, the CD solution of β_j is

$$\widehat{\beta}_j = \frac{S(\sum_{i=1}^n w_i X_{ij} \{Y_i - \widetilde{Y}_i^{(-j)}\}, \lambda_n)}{\sum_{i=1}^n w_i X_{ij}^2}, \tag{2.3}$$

where $\widetilde{Y}_i^{(-j)} = \widetilde{\beta}_0 + \sum_{k:k\neq j} X_{ik}\widetilde{\beta}_k$, and $S(z,\gamma) = \text{sign}(z)(|z|-\gamma)_+$ is called the soft thresholding operator (Donoho and Johnstone, 1994). We can observe that

$$\begin{aligned}\sum_{i=1}^n w_i X_{ij}\{Y_i - \widetilde{Y}_i^{(-j)}\} &= \sum_{i=1}^n w_i X_{ij}(\widetilde{r}_i + X_{ij}\widetilde{\beta}_j) \\ &= \sum_{i=1}^n w_i \widetilde{r}_i X_{ij} + \Big(\sum_{i=1}^n w_i X_{ij}^2\Big)\widetilde{\beta}_j,\end{aligned}$$

thus (2.3) becomes

$$\widehat{\beta}_j = \frac{S(\sum_{i=1}^n w_i \widetilde{r}_i X_{ij} + (\sum_{i=1}^n w_i X_{ij}^2)\widetilde{\beta}_j, \lambda_n)}{\sum_{i=1}^n w_i X_{ij}^2}. \tag{2.4}$$

Then the residual $\widetilde{r}_i$ due to the update in the estimate of β_j is updated by

$$\widehat{r}_i = \widetilde{r}_i + X_{ij}(\widetilde{\beta}_j - \widehat{\beta}_j), \qquad i = 1,\ldots,n.$$

Note that for any given value $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$, the optimal solution $\widehat{\beta}_0$ for the intercept β_0 satisfies

$$\widehat{\beta}_0 = \frac{\sum_{i=1}^n w_i(Y_i - \boldsymbol{X}_i^T\widehat{\boldsymbol{\beta}})}{\sum_{i=1}^n w_i}. \tag{2.5}$$

The solution of (2.2) can be obtained via cycling through the parameters in (2.4)–(2.5) and updating each in turn. As the algorithm completely avoids the large-scale matrix operations, the algorithm enjoys the computational simplicity, speed and stability. Moreover, the computational cost increases only linearly with p, making the algorithm particularly attractive for high-dimensional problems. Issues on the convergence of the CD algorithm can be found in for example, Tseng (2001) and Wu and Lange (2008).

2.1 Extension to Other Penalty Functions

The CD algorithm for the penalized least-squares estimation in (2.2) can be extended in several other ways. One direction is to consider penalty functions instead of the L_1 penalty. For example, consider the optimization problem,

$$\arg\min_{\beta_0,\beta_1,\ldots,\beta_p}\left\{\frac{1}{2}\sum_{i=1}^n w_i(Y_i - \beta_0 - \boldsymbol{X}_i^T\boldsymbol{\beta})^2 + \lambda_n\sum_{j=1}^p t_j|\beta_j|\right\}, \tag{2.6}$$

using the weighted-L_1 penalty, $\lambda_n \sum_{j=1}^{p} t_j|\beta_j|$, where $\{t_j\}$ are non-negative weights associated with the L_1 penalty functions. The above CD procedure continues to work for solving (2.6). The only change to be made is to replace λ_n in (2.4) by $\lambda_n t_j$.

For non-convex penalties, such as the SCAD penalty, $P_\lambda(|\beta|) = \lambda p_\lambda(|\beta|)$, which was proposed by Fan (1997) and whose theoretical properties were demonstrated in Fan and Peng (2004), the first order derivative of $p_\lambda(\beta)$ is given as

$$p'_\lambda(\beta) = \mathrm{I}(\beta \le \lambda) + \frac{(a\lambda - \beta)_+}{(a-1)\lambda}\mathrm{I}(\beta > \lambda), \quad \text{for some } a > 2, \text{ and } \beta > 0,$$

and the penalized estimation is formulated as

$$\arg \min_{\beta_0,\beta_1,\ldots,\beta_p} \left\{ \frac{1}{2}\sum_{i=1}^{n} w_i(Y_i - \beta_0 - \boldsymbol{X}_i^T\boldsymbol{\beta})^2 + \sum_{j=1}^{p} P_{\lambda_n}(|\beta_j|) \right\}. \tag{2.7}$$

In this case, the non-convexity of the SCAD penalty complicates the optimization for (2.7) and the CD algorithm may not be applied similarly; moreover, the convergence of the algorithm may not be guaranteed either. In a special situation where $w_i = 2/n$ and $\sum_{i=1}^{n} X_{ij}^2 = n/2$, the CD iterative solution for β_j has an explicit expression,

$$\widehat{\beta}_j = S^*\Big(\sum_{i=1}^{n} w_i X_{ij}\{Y_i - \widetilde{Y}_i^{(-j)}\}, \lambda_n\Big),$$

where

$$S^*(x,\lambda) = \begin{cases} \mathrm{sign}(x)(|x| - \lambda)_+, & \text{if } |x| \le 2\lambda; \\ \frac{(a-1)x - \mathrm{sign}(x)a\lambda}{(a-2)}, & \text{if } 2\lambda < |x| \le a\lambda; \\ x, & \text{if } |x| > a\lambda, \end{cases}$$

is the SCAD thresholding operator. In general, a local linear approximation (using a Taylor series expansion) can be made to the SCAD penalty. After that, the CD algorithm with the weighted-L_1 penalty can be applied to obtain the solution.

3 Coordinate Descent for Penalized BD Estimation

In this section, we extend the CD algorithm for the penalized least-squares estimation of the linear regression model (2.1) to the general

regression model,

$$E(Y_i \mid \boldsymbol{X}_i) = F^{-1}(\beta_0 + \boldsymbol{X}_i^T \boldsymbol{\beta}), \qquad i = 1, \ldots, n, \tag{3.8}$$

where F is a known link function. For instance, an identity link $F(\mu) = \mu$ corresponds to the linear regression model (2.1); a logit link $F(\mu) = \log(\frac{\mu}{1-\mu})$ is utilized in the logistic regression for binary responses; a log link $F(\mu) = \log(\mu)$ is used in Poisson regression of count responses.

For estimation purpose, the quadratic loss $(y-\mu)^2$, as an error measure used in (2.2), is not suitable for binary responses. Thus, we first discuss a class of loss functions $Q(y, \mu)$ for estimating parameters in model (3.8) with non-Gaussian response variables.

3.1 Bregman Divergence (BD) as the Loss Function

For a given concave function q, Bregman (1967) introduced a device for constructing a bivariate function,

$$Q(\nu, \mu) = -q(\nu) + q(\mu) + (\nu - \mu)q'(\mu). \tag{3.9}$$

Conversely, for a given Q-loss, Zhang, *et al.* (2009) provided necessary and sufficient conditions for Q being a BD, and in the BD case derived an explicit formula for solving the generating q-function. Applying this inverse approach from Q to q, they illustrated that the quadratic function, the (negative) quasi-likelihood function (Wedderburn, 1974; McCullagh, 1983), the Kullback-Leibler divergence (or the deviance loss) for the exponential family of probability functions, and many margin-based loss functions, such as the misclassification loss, the hinge loss for the support vector machine (Vapnik, 1996), the exponential loss used in AdaBoost (Hastie, Tibshirani and Friedman, 2001) are BD.

As an illustration, for a binary response variable, the Bernoulli deviance loss

$$Q(Y, \mu) = -2\{Y \log(\mu) + (1 - Y) \log(1 - \mu)\} \tag{3.10}$$

corresponds to $q(\mu) = -2\{\mu \log(\mu) + (1-\mu) \log(1-\mu)\}$; the Exponential loss

$$Q(Y, \mu) = \exp[-(Y - .5) \log\{\mu/(1 - \mu)\}] \tag{3.11}$$

corresponds to $q(\mu) = 2\{\mu(1 - \mu)\}^{1/2}$.

For another illustration, consider the quasi-likelihood function Q which relaxes the distributional assumption on a random variable Y via the specification, $\partial Q(Y, \mu)/\partial \mu = (Y - \mu)/V(\mu)$, where $\text{var}(Y \mid \boldsymbol{X} =$

$\boldsymbol{x}) = \sigma^2 V(m(\boldsymbol{x}))$ for a nuisance parameter $\sigma^2 > 0$, a known continuous function $V(\cdot) > 0$ and $m(\boldsymbol{x}) = E(Y \mid \boldsymbol{X} = \boldsymbol{x})$. Zhang, *et al.* (2009) verified that the (negative) quasi-likelihood function belongs to the Bregman divergence and derived the generating q-function,

$$q(\mu) = \int_{\mu_0}^{\mu} \frac{s-\mu}{V(s)} ds, \tag{3.12}$$

where μ_0 is a finite constant such that the integral is well-defined.

3.2 Penalized BD Estimation via Coordinate Descent

For the general regression model (3.8), the penalized BD estimation of $(\beta_0, \boldsymbol{\beta})$ can be phrased as

$$\arg \min_{\beta_0,\beta_1,\ldots,\beta_p} \left\{ \frac{1}{n} \sum_{i=1}^{n} Q(Y_i, F^{-1}(\beta_0 + \boldsymbol{X}_i^T \boldsymbol{\beta})) + \lambda_n \sum_{j=1}^{p} t_j |\beta_j| \right\}. \tag{3.13}$$

Penalized BD estimation raises two issues. First, whether the estimator in (3.13) is variable selection consistent and enjoys the oracle property? Second, what is the effective method of solving (3.13)? The first theoretical issue has been investigated in Zhang, *et al.* (2010) for suitably chosen weights $\{t_j\}$, when the dimension p diverges with n at a lower rate. They proposed a penalized componentwise regression (PCR) method for selecting weights,

$$\widehat{t}_j = |\widehat{\beta}_j^{\mathrm{PCR}}|^{-1}, \quad j = 1, \ldots, p,$$

where $\widehat{\beta}_j^{\mathrm{PCR}}$ minimizes the criterion function,

$$\ell_{n,j}^{\mathrm{PCR}}(\beta) = \frac{1}{n} \sum_{i=1}^{n} Q(Y_i, F^{-1}(X_{ij}\beta)) + \kappa_n |\beta|,$$

with some sequence $\kappa_n > 0$. Jiang and Zhang (2010) extended the results to high-dimensional cases where p grows nearly exponentially with n, and demonstrated that the weight selection method via PCR outperforms the componentwise regression (CR) method,

$$\widehat{t}_j = |\widehat{\beta}_j^{\mathrm{CR}}|^{-1}, \quad j = 1, \ldots, p,$$

where $\widehat{\beta}_j^{\mathrm{CR}}$ minimizes the criterion function,

$$\ell_{n,j}^{\mathrm{CR}}(\beta) = \frac{1}{n} \sum_{i=1}^{n} Q(Y_i, F^{-1}(X_{ij}\beta)).$$

For the second issue, Zhang, *et al.* (2010) and Jiang and Zhang (2010) modified the LARS algorithm, but the LARS algorithm is computationally intensive especially when p is large.

In this paper, we intend to explore the CD method for penalized BD estimation. We now describe how to extend the CD algorithm from solving (2.6) to solving (3.13). Note that the major difference between the two criterions arises in the part of loss functions. This motivates us to approximate the part of Q-loss by some weighted form of quadratic loss. We first introduce some necessary notation. Define $\mathrm{q}_j(y;\theta) = (\partial^j/\partial\theta^j)Q(y, F^{-1}(\theta))$, $j = 1, 2$, where $\theta = F(\mu)$. We use C_i or C to denote some generic finite constant. Then by a Taylor series expansion of $Q(Y_i, F^{-1}(\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta}))$ around some initial estimates $\{\widetilde{\beta}_k\}_{k=0}^p$, it follows that

$$\begin{aligned}
& Q(Y_i, F^{-1}(\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})) \\
\approx\ & Q(Y_i, F^{-1}(\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})) \\
& +\mathrm{q}_{1i}\{(\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta}) - (\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})\} \\
& +2^{-1}\mathrm{q}_{2i}\{(\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta}) - (\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})\}^2 \\
=\ & 2^{-1}\mathrm{q}_{2i}\{(\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}) - (\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})\}^2 \\
& -\mathrm{q}_{1i}\{(\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}) - (\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})\} + C_i \\
=\ & 2^{-1}\mathrm{q}_{2i}\Big[\{(\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}) - (\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})\}^2 \\
& -2\mathrm{q}_{1i}/\mathrm{q}_{2i}\{(\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}) - (\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})\} + (\mathrm{q}_{1i}/\mathrm{q}_{2i})^2\Big] + C_i \\
=\ & 2^{-1}\mathrm{q}_{2i}[\{(\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}) - (\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})\} - \mathrm{q}_{1i}/\mathrm{q}_{2i}]^2 + C_i,
\end{aligned}$$

where $\mathrm{q}_{1i} = \mathrm{q}_1(Y_i; \widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})$ and $\mathrm{q}_{2i} = \mathrm{q}_2(Y_i; \widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})$. Hence

$$\frac{1}{n}\sum_{i=1}^n Q(Y_i, F^{-1}(\beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta})) \approx \frac{1}{2}\sum_{i=1}^n \Big(\frac{\widetilde{s}_i}{n}\Big)(\widetilde{Z}_i - \beta_0 - \boldsymbol{X}_i^T\boldsymbol{\beta})^2 + C,$$

where

$$\widetilde{s}_i = \mathrm{q}_2(Y_i; \widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}), \tag{3.14}$$

$$\widetilde{Z}_i = (\widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}) - \frac{\mathrm{q}_1(Y_i; \widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})}{\mathrm{q}_2(Y_i; \widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}})}, \tag{3.15}$$

and C is some constant not depending on the parameters $(\beta_0, \boldsymbol{\beta})$. Thus the minimization problem (3.13) can be approximated by

$$\arg\min_{\beta_0,\beta_1,\ldots,\beta_p} \Big\{\frac{1}{2}\sum_{i=1}^n \widetilde{w}_i(\widetilde{Z}_i - \beta_0 - \boldsymbol{X}_i^T\boldsymbol{\beta})^2 + \lambda_n\sum_{j=1}^p t_j|\beta_j|\Big\}, \tag{3.16}$$

with weights $\widetilde{w}_i = \widetilde{s}_i/n$ and pseudo-responses $\widetilde{Z}_i$, both depending on the parameter values $\{\widetilde{\beta}_k\}$.

The resemblance of (3.16) to (2.6) enables us to employ the CD algorithm to solve (3.16). In practical implementation of (3.16), we need to calculate $\mathrm{q}_2(y;\theta)$ associated with $\widetilde{s}_i$ defined in (3.14) and $\mathrm{q}_1(y;\theta)/\mathrm{q}_2(y;\theta)$ associated with $\widetilde{Z}_i$ defined in (3.15). Examples 3.1–3.3 below illustrate some concrete derivations, where $\widetilde{\theta}_i = \widetilde{\beta}_0 + \boldsymbol{X}_i^T\widetilde{\boldsymbol{\beta}}$.

Example 3.1. For Gaussian response variables, using the quadratic loss $Q(y,\mu) = (y-\mu)^2$ and identity link, we obtain

$$\begin{array}{rcl} \mathrm{q}_1(y;\theta) & = & -2(y-\theta), \quad \mathrm{q}_2(y;\theta) = 2, \\ \dfrac{\mathrm{q}_1(y;\theta)}{\mathrm{q}_2(y;\theta)} & = & -(y-\theta). \end{array}$$

Then

$$\widetilde{s}_i = 2, \qquad \widetilde{Z}_i = \widetilde{\theta}_i + (Y_i - \widetilde{\theta}_i) = Y_i.$$

Example 3.2. For Bernoulli response variables, using the logit link, where $\mu = 1/\{1+\exp(-\theta)\}$, we obtain

- with the deviance loss (3.10),

$$\begin{array}{rcl} \mathrm{q}_1(y;\theta) & = & -2(y-\mu), \quad \mathrm{q}_2(y;\theta) = 2\mu(1-\mu), \\ \dfrac{\mathrm{q}_1(y;\theta)}{\mathrm{q}_2(y;\theta)} & = & -\dfrac{(y-\mu)}{\mu(1-\mu)}. \end{array}$$

  Then for $\widetilde{\mu}_i = 1/\{1+\exp(-\widetilde{\theta}_i)\}$,

$$\widetilde{s}_i = 2\widetilde{\mu}_i(1-\widetilde{\mu}_i), \qquad \widetilde{Z}_i = \widetilde{\theta}_i + \frac{Y_i - \widetilde{\mu}_i}{\widetilde{\mu}_i(1-\widetilde{\mu}_i)}.$$

- With the exponential loss (3.11),

$$\begin{array}{rcl} \mathrm{q}_1(y;\theta) & = & -e^{-(y-1/2)\theta}(y-1/2), \quad \mathrm{q}_2(y;\theta) = e^{-(y-1/2)\theta}/4 > 0, \\ \dfrac{\mathrm{q}_1(y;\theta)}{\mathrm{q}_2(y;\theta)} & = & -4(y-1/2) = 2-4y. \end{array}$$

Then

$$\widetilde{s}_i = \mathrm{q}_2(Y_i;\widetilde{\theta}_i), \qquad \widetilde{Z}_i = \widetilde{\theta}_i - (2-4Y_i).$$

Example 3.3. For count response variables, using the quasi-likelihood generated by the q-function in (3.12) with $V(x) = x$ and log link, we obtain

$$\begin{aligned} \mathrm{q}_1(y;\theta) &= -(y-\mu), \quad \mathrm{q}_2(y;\theta) = \mu, \\ \frac{\mathrm{q}_1(y;\theta)}{\mathrm{q}_2(y;\theta)} &= -\frac{(y-\mu)}{\mu} = 1 - y/\mu, \end{aligned}$$

where $\mu = \exp(\theta)$. Then for $\widetilde{\mu}_i = \exp(\widetilde{\theta}_i)$,

$$\widetilde{s}_i = \widetilde{\mu}_i, \qquad \widetilde{Z}_i = \widetilde{\theta}_i + (1 - Y_i/\widetilde{\mu}_i).$$

Remark 3.1. Note that using the quadratic loss and identity link in Example 3.1, the approximate criterion (3.16) agrees with the target criterion (3.13). For other loss and link functions, (3.16) serves as an approximation to (3.13).

4 Simulation Study

For comparing the speed and accuracy between the CD and LARS algorithms, Wu and Lange (2008) have made the comparison for penalized linear regression with continuous responses, using the convex L_1 and L_2 penalties. Thus for the simulation study in this paper, we will focus on comparing the algorithms for the penalized quasi-likelihood estimation with overdispersed Poisson responses, using both the convex L_1 penalty, weighted-L_1 penalty, and a non-convex SCAD penalty. For each cycle of the CD algorithm, the stopping criterion agrees with that of the LARS algorithm: $\max_{j=0,1,\ldots,p} |\widetilde{\beta}_j^{\text{old}} - \widetilde{\beta}_j^{\text{new}}| < 10^{-3}$, the maximum number of iterations is 40, and $\max_{j=0,1,\ldots,p} |\widetilde{\beta}_j^{\text{old}}| < 10^4$. All computations are performed using Matlab 7.8 on Windows Vista machine with Core 2 duo 3.0 GHz CPU and 4GB memory.

4.1 Overdispersed Poisson Responses

We generate overdispersed Poisson counts Y_i satisfying $\text{var}(Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i) = 2m(\boldsymbol{x}_i)$, where $m(\boldsymbol{x}) = E(Y \mid \boldsymbol{X} = \boldsymbol{x})$. In the predictor $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{ip})^T$, we consider

$$\begin{aligned} &\text{low-dimensional case:} && n = 200,\ p = n/8,\ n/2,\ n-10, \\ &\text{high-dimensional case:} && n = 50,\ p = 2n,\ 4n,\ 8n, \end{aligned}$$

respectively, and take $X_{i1} = i/n - 0.5$. For $j = 2, \ldots, p$, $X_{ij} = \Phi(Z_{ij}) - 0.5$, where Φ is the standard normal distribution function, and $(Z_{i2}, \ldots, Z_{ip})^T \sim N(\mathbf{0}, \rho \mathbf{1}_{p-1}\mathbf{1}_{p-1}^T + (1-\rho)\mathbf{I}_{p-1})$, with $\mathbf{1}_d$ a $d \times 1$ vector of ones and $\mathbf{I}_d$ a $d \times d$ identity matrix. Thus $(X_{i2}, \ldots, X_{ip})$ are marginally Uniform$(-0.5, 0.5)$ random variables and mutually correlated if $\rho \neq 0$. The link function is $\log\{m(\boldsymbol{x})\} = \beta_0 + \boldsymbol{x}^T\boldsymbol{\beta}$, where $\beta_0 = 5$ and $\boldsymbol{\beta} = (2, 2, 0, 0, \ldots, 0)^T$.

For illustrative purpose, 5 procedures for penalized estimators are compared: (I) the SCAD penalty, with an accompanying parameter $a = 3.7$; (II) the L_1 penalty; (III) the weighted-L_1 penalty with weights selected by the CR method; (IV) the weighted-L_1 penalty with weights selected by the PCR method; (V) the oracle estimator using the set of significant variables. For method (I), the SCAD penalty can be locally approximated by a linear function, followed by applying the CD and LARS algorithms. Tables 1-2 summarize the CD and LARS algorithms for the penalized quasi-likelihood estimates of parameters by means of Example 3.3.

Table 1: **(CD algorithm for penalized quasi-likelihood estimation)** *Simulation results, with dependent predictors.* $\rho = 0.2$.

			time			Variable Selection	
(n, p)	TE($\boldsymbol{\beta}$)	Method	(sec)	$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_1$	TE($\widehat{\boldsymbol{\beta}}$)	#CZ (std)	#CNZ (std)
(200, 25)	0.9950	SCAD	2.34	0.1476	1.0391	18.23 (5.4)	3.00 (0.0)
		L_1	1.86	0.2152	1.0566	15.10 (5.2)	3.00 (0.0)
		w. L_1; CR	5.71	0.0908	1.0221	20.75 (3.5)	3.00 (0.0)
		w. L_1; PCR	40.22	0.0853	1.0199	20.91 (3.5)	3.00 (0.0)
		Oracle	0.08	0.0522	1.0129	23.00	3.00
(200, 100)	1.0108	SCAD	12.76	0.1719	1.0586	93.45 (10.3)	3.00 (0.0)
		L_1	10.91	0.2962	1.1007	88.16 (14.4)	3.00 (0.0)
		w. L_1; CR	25.90	0.0968	1.0338	95.86 (3.5)	3.00 (0.0)
		w. L_1; PCR	168.56	0.0817	1.0306	96.07 (3.6)	3.00 (0.0)
		Oracle	0.07	0.0503	1.0250	98.00	3.00
(200, 190)	1.0144	SCAD	65.58	0.1777	1.0705	183.08 (11.7)	3.00 (0.0)
		L_1	52.93	0.2842	1.1218	179.74 (15.3)	3.00 (0.0)
		w. L_1; CR	87.20	0.1029	1.0406	185.17 (3.6)	3.00 (0.0)
		w. L_1; PCR	524.69	0.0889	1.0345	185.52 (4.3)	3.00 (0.0)
		Oracle	0.07	0.0482	1.0279	188.00	3.00
(50, 100)	1.0108	SCAD	17.70	0.3985	1.2182	89.18 (3.9)	3.00 (0.0)
		L_1	12.37	0.5285	1.3324	87.68 (5.0)	3.00 (0.0)
		w. L_1; CR	28.56	0.3445	1.2449	93.02 (6.0)	3.00 (0.0)
		w. L_1; PCR	225.55	0.2923	1.1943	93.70 (5.9)	3.00 (0.0)
		Oracle	0.06	0.0951	1.0698	98.00	3.00
(50, 200)	0.9947	SCAD	21.89	0.5514	1.2737	185.17 (7.2)	3.00 (0.0)
		L_1	15.56	0.7525	1.4835	182.24 (12.1)	3.00 (0.0)
		w. L_1; CR	42.77	0.4502	1.5213	192.68 (8.9)	3.00 (0.0)
		w. L_1; PCR	325.25	0.3442	1.2159	193.06 (7.8)	3.00 (0.0)
		Oracle	0.06	0.1044	1.0576	198.00	3.00
(50, 400)	1.013	SCAD	34.24	0.6773	1.3493	379.66 (9.9)	3.00 (0.0)
		L_1	24.83	0.8532	1.6077	378.21 (13.6)	3.00 (0.0)
		w. L_1; CR	73.35	0.5832	2.2824	390.39 (9.1)	2.98 (0.1)
		w. L_1; PCR	548.72	0.4918	2.1782	391.99 (7.0)	2.98 (0.1)
		Oracle	0.06	0.1029	1.0789	398.00	3.00

Table 2: **(LARS algorithm for penalized quasi-likelihood estimation)** *Simulation results, with dependent predictors.* $\rho = 0.2$.

			time			Variable Selection	
(n, p)	TE($\boldsymbol{\beta}$)	Method	(sec)	$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_1$	TE($\widehat{\boldsymbol{\beta}}$)	#CZ (std)	#CNZ (std)
(200, 25)	0.9950	SCAD	18.20	0.1476	1.0390	17.86 (5.7)	3.00 (0.0)
		L_1	19.84	0.2147	1.0563	14.71 (5.4)	3.00 (0.0)
		w. L_1; CR	24.61	0.0908	1.0221	20.67 (3.6)	3.00 (0.0)
		w. L_1; PCR	173.05	0.0859	1.0200	20.74 (3.8)	3.00 (0.0)
		Oracle	0.08	0.0522	1.0129	23.00	3.00
(200, 100)	1.0108	SCAD	118.72	0.1715	1.0584	93.15 (10.6)	3.00 (0.0)
		L_1	129.72	0.2963	1.1006	87.55 (15.2)	3.00 (0.0)
		w. L_1; CR	164.14	0.0968	1.0338	95.68 (3.8)	3.00 (0.0)
		w. L_1; PCR	1024.78	0.0804	1.0304	96.00 (3.8)	3.00 (0.0)
		Oracle	0.08	0.0503	1.0250	98.00	3.00
(200, 190)	1.0144	SCAD	1281.72	0.1774	1.0703	182.68 (12.4)	3.00 (0.0)
		L_1	1170.75	0.2844	1.1217	179.15 (16.2)	3.00 (0.0)
		w. L_1; CR	1358.57	0.1052	1.0409	184.81 (4.2)	3.00 (0.0)
		w. L_1; PCR	6707.61	0.0901	1.0348	185.32 (4.5)	3.00 (0.0)
		Oracle	0.07	0.0482	1.0279	188.00	3.00
(50, 100)	1.0108	SCAD	362.40	0.3980	1.2177	88.80 (3.9)	3.00 (0.0)
		L_1	216.17	0.5209	1.3299	87.39 (4.8)	3.00 (0.0)
		w. L_1; CR	282.21	0.3447	1.2452	92.87 (6.0)	3.00 (0.0)
		w. L_1; PCR	2227.88	0.2943	1.1950	93.44 (6.2)	3.00 (0.0)
		Oracle	0.06	0.0951	1.0698	98.00	3.00
(50, 200)	0.9947	SCAD	481.37	0.5720	1.2868	183.97 (7.9)	3.00 (0.0)
		L_1	265.61	0.7284	1.4690	182.46 (7.8)	3.00 (0.0)
		w. L_1; CR	300.29	0.4372	1.5113	192.79 (6.8)	3.00 (0.0)
		w. L_1; PCR	2460.30	0.3463	1.2276	193.06 (6.4)	3.00 (0.0)
		Oracle	0.06	0.1044	1.0576	198.00	3.00
(50, 400)	1.013	SCAD	602.67	0.6879	1.3522	378.71 (9.7)	3.00 (0.0)
		L_1	320.28	0.8240	1.6084	379.61 (8.6)	3.00 (0.0)
		w. L_1; CR	367.40	0.5844	2.2840	390.25 (8.9)	2.98 (0.1)
		w. L_1; PCR	3009.60	0.5196	1.9658	391.28 (8.3)	2.99 (0.1)
		Oracle	0.06	0.1029	1.0789	398.00	3.00

First, to examine the effect of penalized regression estimates on parameter estimation, we generate 100 training sets of size n. The tuning constants λ_n for the training set in each simulation for methods (I)–(II) are selected via a grid search separately to minimize the (negative) quasi-likelihood on a test set of size equal to that of the training set; λ_n and κ_n for methods (III) and (IV) are searched on a surface of grid points. For each training set, the test error (TE) is calculated by $\sum_{\ell=1}^{L} Q(y_\ell, \widehat{m}(\boldsymbol{x}_\ell))/L$, at a sequence $\{(\boldsymbol{x}_\ell, y_\ell)\}_{\ell=1}^{L=5000}$ randomly generated. Columns titled TE($\boldsymbol{\beta}$) and TE($\widehat{\boldsymbol{\beta}}$) refer to the TE calculated using the true and estimated parameters. To further assess the accuracy of the penalized estimates, the average of $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_1$ across those 100 training sets is obtained in the 5th column. It is clearly seen that if the true model coefficients are sparse, the penalized estimators preform reasonably well.

Second, to study the utility of penalized estimators in revealing the effects in variable selection under quasi-likelihood, Tables 1-2 list a column titled CZ on the average total number of coefficients which are correctly estimated to be zero when the true coefficients are zero, and a

column titled CNZ on the average total number of coefficients which are correctly estimated to be nonzero when the true coefficients are nonzero. The standard deviations of the corresponding estimations across training sets are given in brackets. Overall, the penalized estimators help yield a sparse solution and build a parsimonious model.

In summary, the CD and LARS algorithms offer comparable performance in sparsity recovery. But the former gains computational superiority. See the column titled "time" (in seconds) which gives the total time of each penalized method in each combination of (n, p). The SCAD and weighted-L_1 penalties outperform the L_1 in terms of regression estimation and variable selection. As expected, the oracle estimator, which is practically infeasible optimal, performs better than the other four penalized estimators.

5 Real Data Application

To further illustrate the usefulness of the CD and LARS algorithms in penalization BD for regression and classification, we consider the Arrhythmia Data Set, which is publicly available at the UCI Machine Learning Repository; see
`http://archive.ics.uci.edu/ml/datasets/Arrhythmia`.

The Arrhythmia dataset (Güvenir, *et al.*, 1997) consists of 452 patient records in the diagnosis of cardiac arrhythmia. Each record contains 279 clinical measurements, from electrocardiography signals and some other information such as sex, age, and weight, along with the decision of an expert cardiologist. In the data, class 01 refers to normal electrocardiography, class 02–class 15 each refer to a particular type of arrhythmia, and class 16 refers to the unclassified rest.

We intend to predict whether a patient can be categorized as either normal or abnormal electrocardiography. After deleting missing values and class 16, the remaining 430 patients with 257 attributes are used in the classification. To evaluate the performance of the penalized estimates of model parameters in $\text{logit}\{P(Y = 1 \mid X_1, \ldots, X_{257})\} = \beta_0 + \sum_{j=1}^{257} \beta_j X_j$, we randomly split the data into a training set and a test set in the ratio 2 : 1. For each training set, the tuning constant is selected by minimizing a 3-fold cross validated estimate of the misclassification rate; λ_n and κ_n for the penalized componentwise regression are searched on a surface of grid points. We calculate MMR, the mean of the misclassification rates and the average number of selected variables over 100 random splits. Results using the CD and LARS algorithms are given

in Table 3 and Table 4 respectively. Again, both algorithms deliver comparable results, and the CD algorithm (about 1 hour) is faster than the LARS algorithm (about 38 hours). It is seen that the penalized classifier using the deviance loss and that using the exponential loss have similar values of misclassification rates. In contrast, the non-penalized classifiers select all attributes, yielding much higher misclassification rates.

Table 3: **(CD algorithm)** *Arrhythmia data: Mean misclassification rate and the average number of selected variables.* MMR*: mean of the misclassification rates.*

Loss	Method	MMR	# Selected Variables
Deviance	SCAD	0.2506	13.96
	L_1	0.2363	43.65
	weighted L_1; CR	0.2187	39.09
	weighted L_1; PCR	0.2243	25.05
	non-penalized	0.4057	257.00
Exponential	SCAD	0.2624	12.98
	L_1	0.2361	41.87
	weighted L_1; CR	0.2151	34.77
	weighted L_1; PCR	0.2274	17.33
	non-penalized	0.4326	257.00

Table 4: **(LARS algorithm)** *Arrhythmia data: Mean misclassification rate and the average number of selected variables.* MMR*: mean of the misclassification rates.*

Loss	Method	MMR	# Selected Variables
Deviance	SCAD	0.2560	19.40
	L_1	0.2366	43.89
	weighted L_1; CR	0.2304	40.11
	weighted L_1; PCR	0.2299	27.53
	non-penalized	0.4057	257.00
Exponential	SCAD	0.2639	14.27
	L_1	0.2410	43.52
	weighted L_1; CR	0.2310	39.03
	weighted L_1; PCR	0.2330	20.43
	non-penalized	0.4326	257.00

References

Brègman, L. M. (1967), A relaxation method of finding a common point of convex sets and its application to the solution of problems in convex programming. U.S.S.R. Comput. Math. and Math. Phys., **7**, 620–631.

Breiman, L. (1995), Better subset regression using the nonnegative garrote. Technometrics, **37**, 373–384.

Donoho, D. L. and Johnstone, J. M. (1994), Ideal spatial adaptation by wavelet shrinkage. Biometrika, **81**, 425-455.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), Least angle regression. Ann. Statist., **32**, 407–499.

Fan, J. (1997), Comments on "Wavelets in statistics: a review," by A. Antoniadis. J. Italian Statist. Soc., **6**, 131–138.

Fan, J. and Li, R. (2001), Variable selection via nonconcave penalized likelihood and its oracle properties. J. Amer. Statist. Assoc., **96**, 1348–1360.

Fan, J. and Peng, H. (2004), Nonconcave penalized likelihood with a diverging number of parameters. Ann. Statist., **32**, 928–961.

Frank. I. E. and Friedman. J. H. (1993), A statistical view of some chemometrics tools. Technometrics, **35**, 109–135.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007), Pathwise coordinate optimization. Ann. Appl. Statist., **1**, 302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, January 2010, volume 33, issue 1.

Güvenir, H. A., Acar, B., Demiröz, G., and Çekin, A. (1997), A supervised machine learning algorithm for arrhythmia analysis. Computers in Cardiology, **24**, 433–436.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), The Elements of Statistical Learning. Springer.

Jiang, Y. and Zhang, C. M. (2010), High-dimensional regression and classification under a class of convex loss functions. Technical report #1149, Dept. of Statistics, University of Wisconsin-Madison.

McCullagh, P. (1983), Quasi-likelihood functions. Ann. Statist., **11**, 59–67.

Tibshirani, R. (1996), Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc. Ser. B, **58**, 267–288.

Tseng, P. (2001), Convergence of block coordinate descent method for nondifferentiable maximization. J. Optim. Theory Appl., **109**, 473–492.

Vapnik, V. (1996), The Nature of Statistical Learning Theory. New York: Springer-Verlag.

Wedderburn, R. W. M. (1974), Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. Biometrika, **61**, 439–447.

Wu, T. T. and Lange, K. (2008), Coordinate descent algorithms for Lasso penalized regression. Annals of Applied Statistics, **1**, 224–244.

Zhang, C. M., Jiang, Y., and Shang, Z. (2009), New aspects of Bregman divergence in regression and classification with parametric and nonparametric estimation. Canad. J. Statist., **37**, 119–139.

Zhang, C.M., Jiang, Y., and Chai, Y. (2010), Penalized Bregman divergence for large-dimensional regression and classification. Biometrika, **97**, 551–566.